Молдавский государственный университет

Факультет математики и информатики

Департамент информатики

**Чобану Артём**

**Разработка Веб - приложений на основе**

**Микросервисной Архитектуры**

**0613.4 ИНФОРМАТИКА**

Дипломная работа

Директор департамента: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Капчеля Титу, др. конф.

Научный руководитель: **\_\_\_\_\_\_\_\_\_\_\_\_**\_\_ Латул Георгий, преподаватель.

Автор: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ ***Чобану Артём***

Кишинэу – 2022

Содержание

[Ключевые слова и сокращения 3](#_Toc93750343)

[Введение 4](#_Toc93750344)

[Микросервисная Архитектура. 5](#_Toc93750345)

[Примеры использования и паттерны 6](#_Toc93750346)

[Внутренняя архитектура каждого микросервиса. 7](#_Toc93750347)

[Проектирование архитектуры приложения. 8](#_Toc93750348)

[Реализация проекта. 9](#_Toc93750349)

[Заключение 10](#_Toc93750350)

[Литература и документация: 11](#_Toc93750351)

А Н Н О Т А Ц И Я

На дипломную работу

„**Разработка Веб-приложений на основе Микросервисной Архитектуры**”

**Ключевые слова:** Веб-Сервис, Сервис-Ориентированная Архитектура, Микросервис, Микросервисная Архитектура, Монолит.

Целью работы является изучение Микросервисной Архитектуры выявление её преимуществ и недостатков.

В дипломной работе проводится сравнение Микросервисной и Монолотной Архитектур, приводятся примеры шаблонов проектированя, лучших практик.

Для примера был разработан проект чата, состоящего из двух микросервисов. Проект разработан на платформе ASP.NET 6.0 с использованием языка программирования C# 10.0, что означает, что проект является кросплатформенным.

В работе будет представлена архитектура, части кода, а также скриншоты проекта, примеры его работы.

Дипломная работа состоит из введения, \_ глав и приложения. В первой главе рассматриваются \_\_\_\_. Во второй главе рассмотрен \_\_\_. Третья и четвертая главы посвящены проектированию и разработке приложения \_\_\_.

Дипломная работа состоит из \_\_ страниц и содержит \_\_ таблицу, \_\_ рисунков, \_\_ схему, \_\_ страниц приложений.

# Ключевые слова и сокращения

Веб-сервис, Сервис-Ориентированная Архитектура, Микросервис, Микросервисная Архитектура, Vertical Slice Архитектура, Message Queuing Service (Служба очереди сообщений), API Gateway (API Шлюз), Монолит.

CQRS – Command Query Responsibility Segregation

MQS – Message Queueing Service

SOA – Service Oriented Architecture

API – Application Programming Interface

# Введение

Современные веб-приложения становятся всё сложнее, требования растут, и касаются множества аспектов, базирующихся на предыдущем опыте, и необходимости для конкретного проекта. Ниже приведены основные требования к комплексным веб-приложениям:

1. Отказоустойчивость -

…

1. Простота развёртывания –

…

1. Масштабирование –

…

1. Заменяемость –

…

1. Простота тестирования –

…

1. Количество зависимостей –

…

1. Безопасность –

…

# Микросервисная Архитектура.

Веб-Сервис – это …

Сервис-Ориентированная Архитектура – это…

Монолитная Архитектура – это…

Микросервисная Архитектура – это…

Сравнение Микросервисной и Монолитной Архитектур (таблица). Сложность разработки, сложность внесения изменений, заменяемость, развёртывание, сложность поддержки, отказоустойчивость, тестирование, масштабирование

Независимость микросервисов, противоречия: микросервисы должны общаться, но при этом быть как можно более независимыми).

Разнообразие используемых технологий для создания каждого микросервиса.

Возможности, доступные при использовании облачных технологий.

Балансировка нагрузки, возможность запускать несколько экземпляров(instance) каждого микросервиса.

Logging(журналирование) при использовании микросервисов, наличие централизованного хранилища логов.

# Примеры использования и паттерны

Паттерн API Gateway и примеры (паттерн может быть использован во многих вариантах в зависимости от задач проекта)

Авторизация (в том числе в виде Gateway)

Паттерн API Decomposition – это…

Использование Message Queuing Сервисов (на примере RabbitMQ и BizTalk) для общения микросервисов, а также для инициации событий (web job).

Примеры проблем, возникающих при использовании Микросервисной Архитектуры.

# Внутренняя архитектура каждого микросервиса.

Vertical Slice Архитектура, CQRS, возможность разделения ответственностей на Read/Write.

# Проектирование архитектуры приложения.

Описание требований, задач, функционала.

Описание двух микросервисов, общающихся через RabbitMQ.

Графическое представление архитектуры приложения

Архитектура каждого микросервиса в отдельности.

# Реализация проекта.

Скриншоты, демонстрация функций.

# Заключение

Короткий повтор преимуществ и недостаткой, а также описание проектов, для которых лучше подходит микросервисная или монолитная архитектура.

# Литература и документация:

\*ссылки, документация и книги\*